

Status and developments for DAQ system MBS v6.3*

J. Adamczewski-Musch¹, N. Kurz¹, S. Linev¹, and the FAIR@GSI project¹

¹GSI, Darmstadt, Germany

Introduction

The software framework MBS (Multi Branch System) has been established as common DAQ system at GSI for more than 20 years [1]. Various experiments apply MBS for data taking. Besides supporting the users when setting up the experiment, the MBS framework is continuously being improved and developed due to changed requirements. This article reports the most recent experiences and ongoing developments of MBS.

MBS installation update

In 2014 19 new MBS systems (12 VME based, 7 PC/PCI Express based) have been installed and came into operation at GSI and worldwide. In addition 9 MBS systems have been upgraded to the MBS production version or to new operating system versions. Furthermore three travel MBS systems for experiments outside of GSI have been set up.

Developments

Improved PCI Express (PCIe) readout

The MBS driver software for PEXOR/KINPEX PCIe optical receiver boards has been improved for x86 Linux platforms [2]. This consists in a new kernel module *mbspex* with corresponding C library. Additionally, a command line tool *gospicmd* allows front-end configuration and controls concurrently to running MBS DAQ. If the MBS user readout function calls *mbspex* library functions only, a safe read-out will be possible simultaneous with external control system access. However, backward compatibility to existing readout code is provided by the new driver.

White rabbit timing system

In the GSI test beam campaign 2014 synchronisation with White Rabbit timing receivers (WTR) [3] has been introduced successfully at the experiment Land/R3B for the first time. With the help of the VETAR (VME) and PEXARIA (PCI Express) WTRs several independent MBS systems running with global triggers systems have been synchronised.

In addition a prototype of the R3B silicon tracker (SITR) could be included in the White Rabbit time stamp scheme. The SITR is designed to run without global, but with individual detector input signal triggers (local triggers). It provides for each signal over threshold a charge (energy) and a White Rabbit time stamp.

To accomplish synchronisation with the MBS systems running with global triggers, two LVDS signal lines have been sent from a VETAR WTR into the SITR: 1) White Rabbit controlled 200 MHz for time counting; 2) A 100 KHz pulse train, phase stable with the 200 MHz clock. The 64 bit White Rabbit time reference is sent serially encoded in each ten microsecond interval of consecutive pulses.

MBS systems running with global triggers and a system running with local triggers could be successfully synchronised. This serves as blueprint for FAIR data acquisition setups, where these conditions will be standard.

Controls with DABCs

Control and monitoring MBS nodes via tcp/ip socket from remote is since long time possible: a status server would deliver run state and rate information to monitoring clients like Go4, and multiple MBS nodes would receive commands via prompter socket. Additionally, for DABC controls two new channels have been introduced for MBS v6.3: a remote log server (“logrem”) sending log output to the control client; and a remote command channel (“cmdrem”) in the MBS dispatcher process that inserts client commands beside the local MBS console input. These new channels may be started from the beginning by command line argument (“*mbs -dabc*”), or started from running dispatcher session by new commands (“*start logrem*”, “*start cmdrem*”). DABC library offers modules to connect these channels. The DABC executable *mbscmd* can either control the MBS dispatcher process via *cmdrem* socket, or the MBS prompter process via the already existing multi-node control socket.

Webserver GUI

The MBS control sockets may be accessed by a separate DABC process running an http web server [4]. The DABC web server will forward http command requests via the *cmdrem* channel to MBS dispatcher. It will retrieve log messages from MBS *logrem* channel and DAQ status from MBS status socket, and may export this information to the web. This allows remote control of each single node dispatcher from web browser. In addition to the default DABC tree display of these web objects, a dedicated MBS control GUI has been developed for this DABC web server using JavaScript with the *jQuery UI* environment.

* PSP codes: SFRS 2.4, ILIMA 1.2.2.4, R3B 1.2.5.1.4, 1.2.6.6, 1.2.5.1.2.5, SPARC 1.3.1.3.11., SD-SEM 2.3.6.5.1.6.30



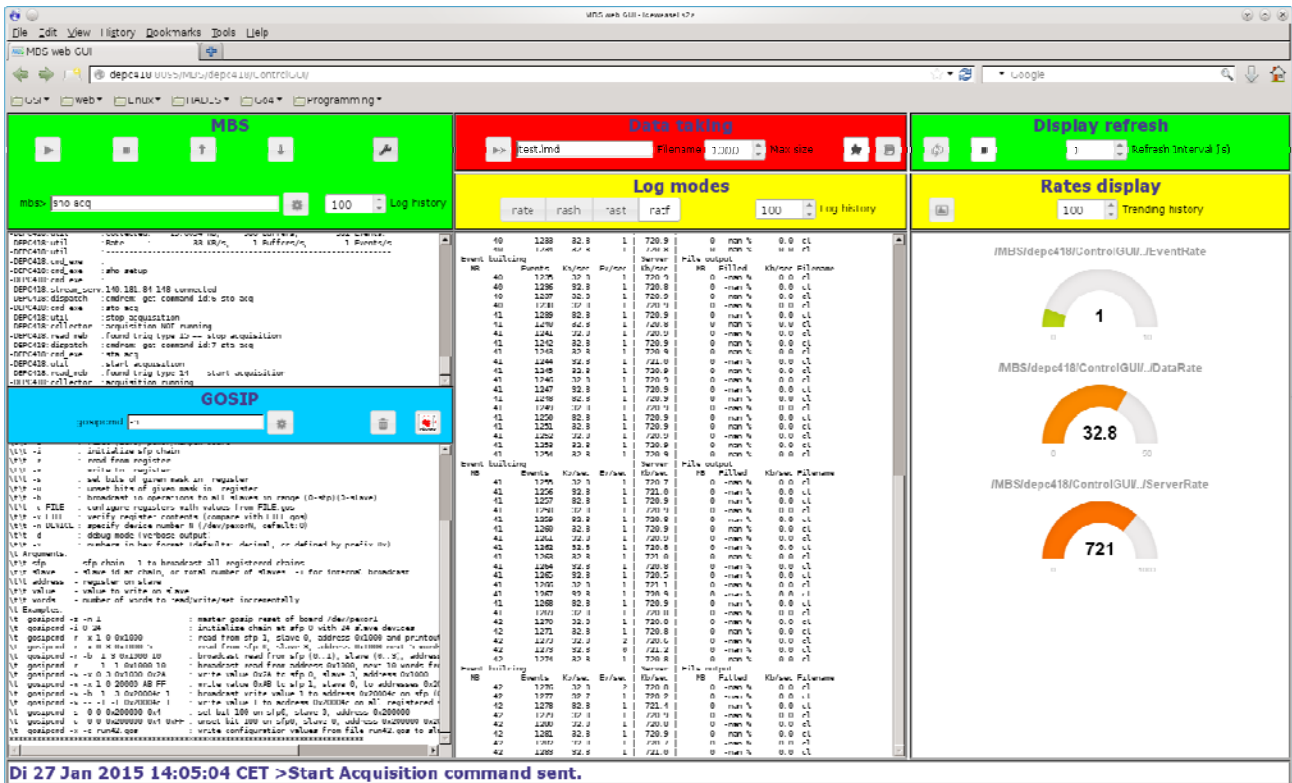


Figure 1: The MBS web browser control GUI

Figure 1 shows a screenshot of such a GUI. The dispatcher interactive command line appears in the browser and offers additional shortcut buttons for mostly used commands, like start/stop acquisition, *@startup*, *@shutdown*. Data taking commands and status is shown in a separate tile with control buttons for open/close file, file mode and RFIO/local disk set up. The run or file status of the acquisition is visualized by red or green background colours of the control tiles. Data rates are displayed both as graphical gauge or trending rate meters, and as text terminal output of the previously known *rate*, *rash*, *rast*, or *ratf* shell commands of MBS. Automatic refresh frequency of the browser display can be set in a dedicated box. Moreover, when running on MBS Linux PC, the DABC web server also offers to call the frontend configuration tool *gosipcmd* [2], so a command line and output box for this tool can optionally be shown in the browser. Additionally, a dedicated frontend configuration GUI, like POLAND setup [2], may be started by a button in a different browser window.

Outlook

The new features described above are partially already installed at GSI as development version 6.x (“mbslogin deve”) and can be tested. They will be part of the future production version MBS 6.3 to be released in 2015. Moreover, adjustments to new PC hardware and 64bit Linux are currently under work and will be available with this release.

References

- [1] Multi Branch System home page: <http://www.gsi.de/mbs>
- [2] J. Adamczewski-Musch, N. Kurz, S. Linev, “mbspex driver software for PEXOR/KINPEX readout boards”, this report
- [3] D. Beck et al., “Paving the Way for the FAIR General Machine Timing System”, GSI scientific report 2013, doi:0.15120/GR-2014-1-FG-CS-08
- [4] S. Linev, “Web interface in DABC”, GSI scientific report 2013, doi:10.15120/GR-2014-1-FG-CS-12

